Text Color Buttons: README

The default Unity buttons allow to specify colors that are applied to the target graphic depending on the button state. Most buttons have text labels, and sometimes you also want to change the text color depending on the button state (or change only the text color and keep the background graphics unaltered).

This is what `TextColorButtons` allows you to do, by extending the Unity `Button` class to handle an additional set of colors for the text label.

Includes two variants, for buttons using as labels `UnityEngine.UI.Text` or `TMPro.TMP_Text` components.

Also provided some extension methods to easily setup colors by code.

Getting Started

If you want to add the functionality to an existing button:

- remove the `Button` script from the button
- add the `TextColorButton` script
- configure the text color properties in the inspector

If you want to add a new button to a `Canvas`:

- drag to the Scene window one of the TextColorButtons prefabs
    - TextColorButton is a Button with a Text label
    - TMP_TextColorButton is a Button with a TMP_Text label
- configure the text color properties in the inspector

Samples

The featured sample scene, scn_TextColorButtons.unity, shows some simple usage examples. In Demo\Prefabs there are four buttons using the TextColorButtons scripts, one for each kind of button used in the demo. We suggest to similarly setup prefabs for the different kinds of buttons needed in your project, setting the right graphics/color settings only once. Unless you prefer to setup components via code, of course.

Techical details and caveats

The provided scripts try to extend the default Unity buttons ergonomically. The fade logic is replicated from the Unity implementation, so the text color fading behaviour should match the target graphics fading behaviour.

The TextColorBlock closely matches the ColorBlock class, but keeps open the possibility to extend it if needed (with more colors/fading options, e.g. using an easing function and not linear interpolation). We initially tried using another ColorBlock instance for m_TextColors, but the Unity inspector didn’t quite like that, so we opted for adding the TextColorBlock class.

All the core logic is in the ATextColorButton generic abstract class, and the two derived classes TextColorButton and TMP_TextColorButton implement the setTextColor()/getTextColor() methods for, respectively, buttons having a UnityEngine.UI.Text label or a TMPro.TMP_Text label.

The same pattern is used for the Editor classes.

Remember that to set colors by code, you should set the whole ColorBlock or TextColorBlock structs:

```
[SerializeField] TextColorButton m_rTCB; // reference to a TextColorbutton

void Start() {
  // this works
  TextColorBlock tcb = m_rTCB.textColors;
  tcb.textNormalColor = Color.black;
```

```
   m_rTCB.textColors = tcb;

   // this code is not valid
   // m_rTCB.textColors.textNormalColor = Color.black;
  }
```

If you don’t like that, consider adding to your code `using BinaryCharm.UI.TextColorButtons.Extensions;` This will enable you to use the extension methods provided by the two static classes `ButtonExtensions` and `TextoColorButtonExtensions`:

```
  [SerializeField] TextColorButton m_rTCB; // reference to a TextColorbutton

  void Start() {
   m_rTCB.setNormalTextColor(Color.black);

   // setNormalTextColor is defined in
   // BinaryCharm.UI.TextColorButtons.Extensions.TextoColorButtonExtensions
  }
```

API overview

The `TextColorButton` and `TMP_TextColorButton` scripts try to expand ergonomically `Button` and need no API documentation. Just access `TextColorBlock` with `.textColors` like you access `ColorBlock` with `.colors`.

The extension methos provided by `ButtonExtensions` and `TextColorButtonExtensions` offer utility methods to easily modify a single value of the `colors` and `textColors` properties. Their usage is pretty obvious too.

- `ButtonExtensions` methods to set `color` members
  - `void setGfxColorFadeDuration(float fDurationSecs)`
  - `void setGfxColorMultiplier(float fColorMultiplier)`
  - `void setNormalGfxColor(Color c)`
  - `void setHighlightedGfxColor(Color c)`
  - `void setPressedGfxColor(Color c)`
  - `void setSelectedGfxColor(Color c)`
  - `void setDisabledGfxColor(Color c)`

- TextColorButtonExtensions: methods to set textColors members
    - void setTextColorFadeDuration(float fDurationSecs)
    - void setTextColorMultiplier(float fColorMultiplier)
    - void setNormalTextColor(Color c)
    - void setHighlightedTextColor(Color c)
    - void setPressedTextColor(Color c)
    - void setSelectedTextColor(Color c)
    - void setDisabledTextColor(Color c)

Contact us

You can find out all the ways to contact us on https://www.binarycharm.com.
If you need technical support about this component, please write to support@binarycharm.com by specifying [Text Color Buttons] in the subject.